

200

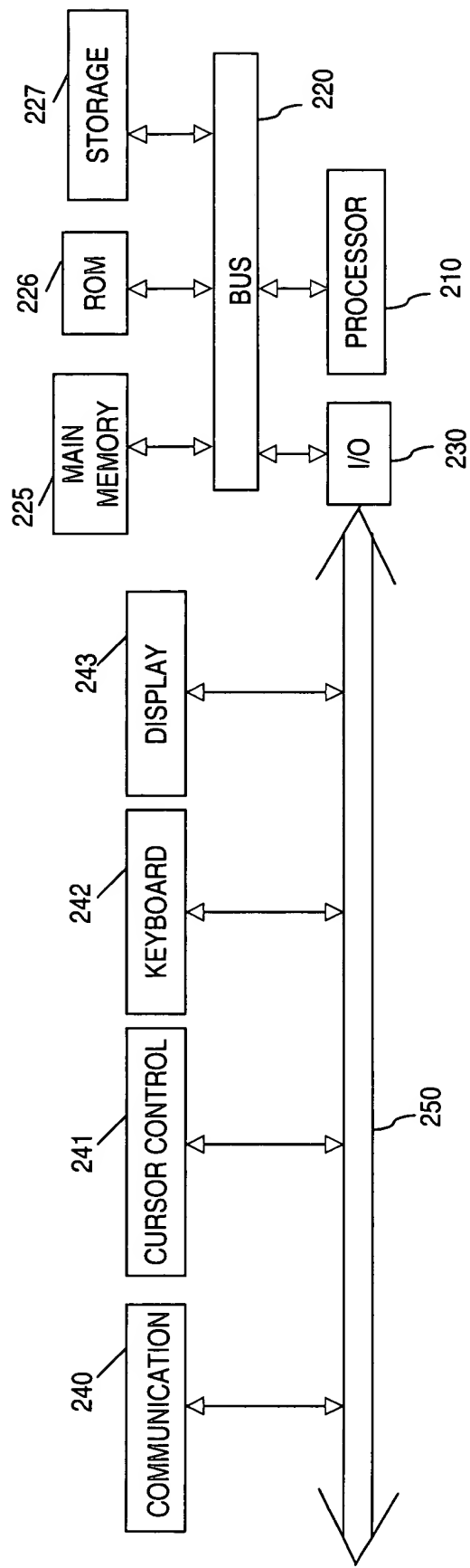


FIG. 1

FIG. 2 OF 2

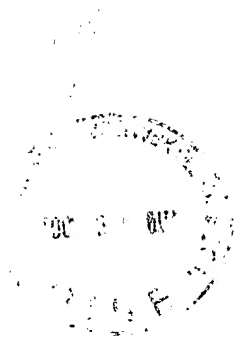
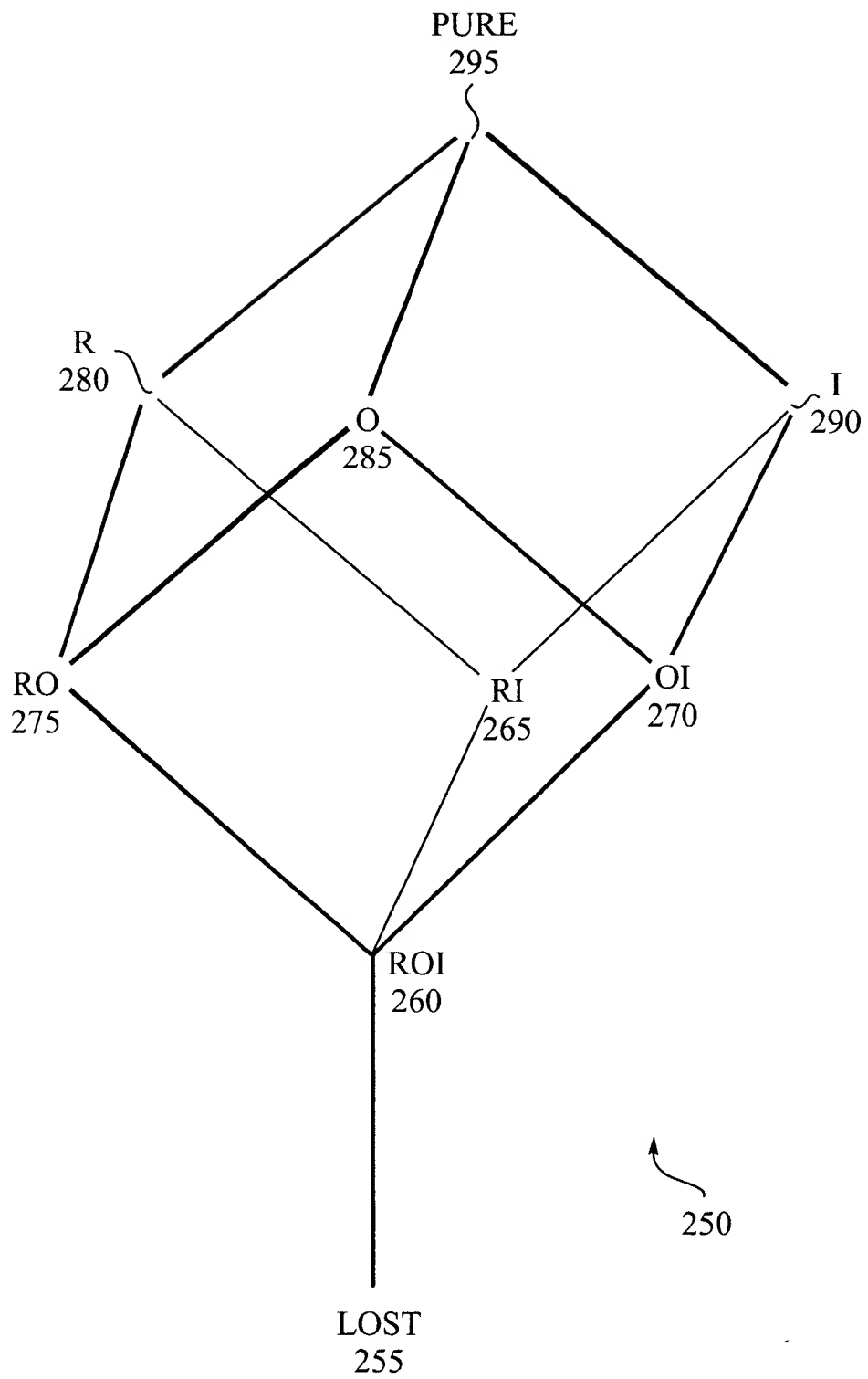


Fig. 2

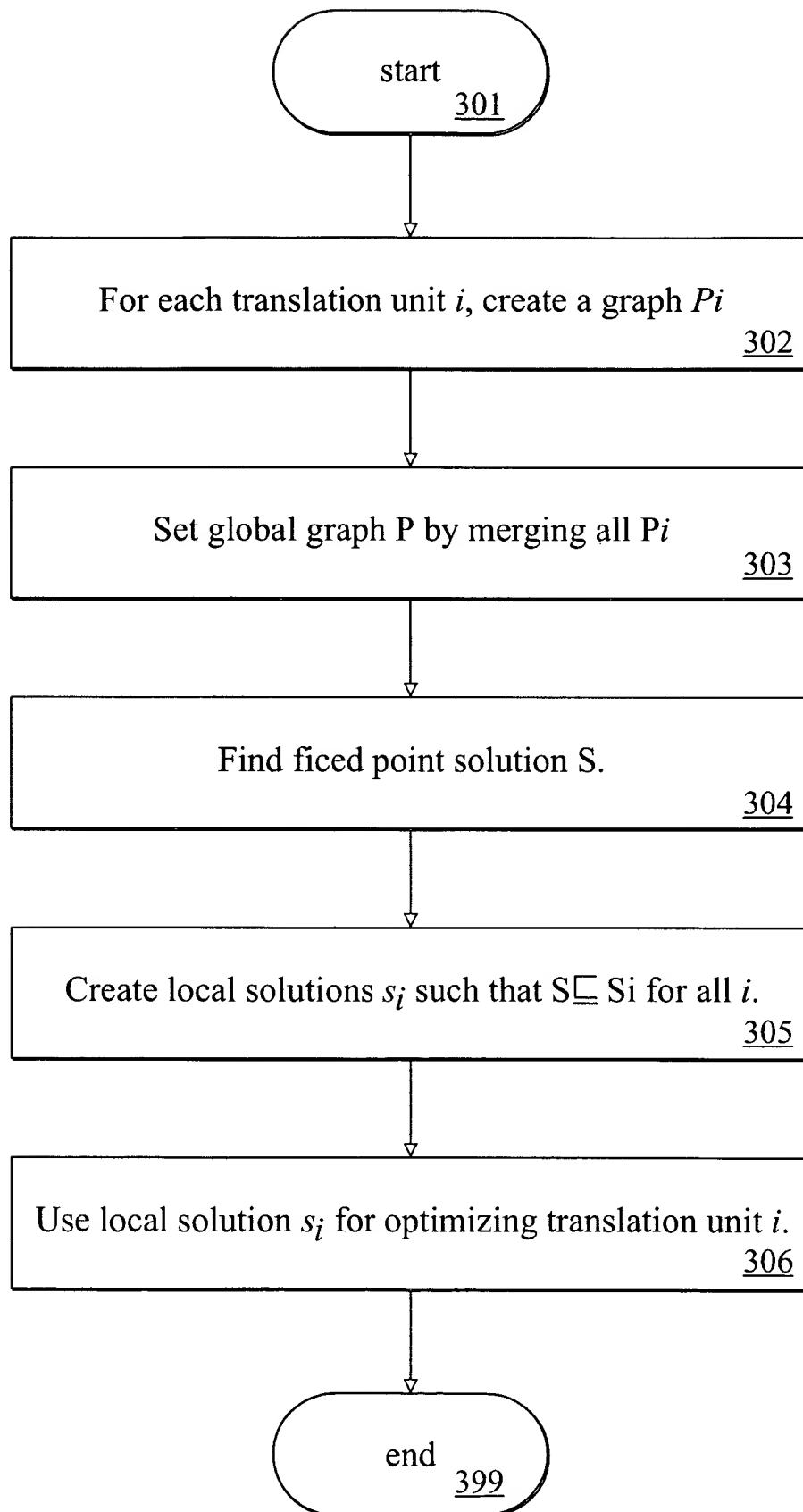


Fig. 3

TOP OF PAGE



Fujnction	Fujnction (x,y)
410 \sim TOP	(PURE,PURE)
420 \sim COPY	(y,y)
430 \sim IN_TO_LOST	if $y \leq I \Rightarrow$ (LOST,LOST) otherwise \Rightarrow (PURE,PURE)
440 \sim UNRETURN	
450 \sim COPY_AND_IN_TO_LOST	if $y = \text{LOST} \Rightarrow$ (LOST,LOST) otherwise \Rightarrow (z,z) where $z = y \sqcup OI$
460 \sim CAT_FORMAL	if $y \leq I \Rightarrow$ (LOST,LOST) otherwise \Rightarrow (y,y)
470 \sim CAT_ACTUAL	(y,PURE)
	(PURE,y)
480 \sim GATE	if $x = \text{LOST} \Rightarrow$ (LOST,LOST) else if $x < R$ (z,z) where $z = (x \sqcup OI) \sqcap y$ else (z,z) where $z = (x \sqcup OI)$

Fig. 4A

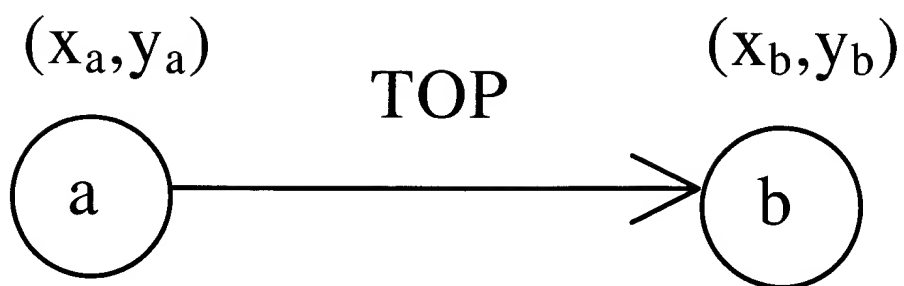
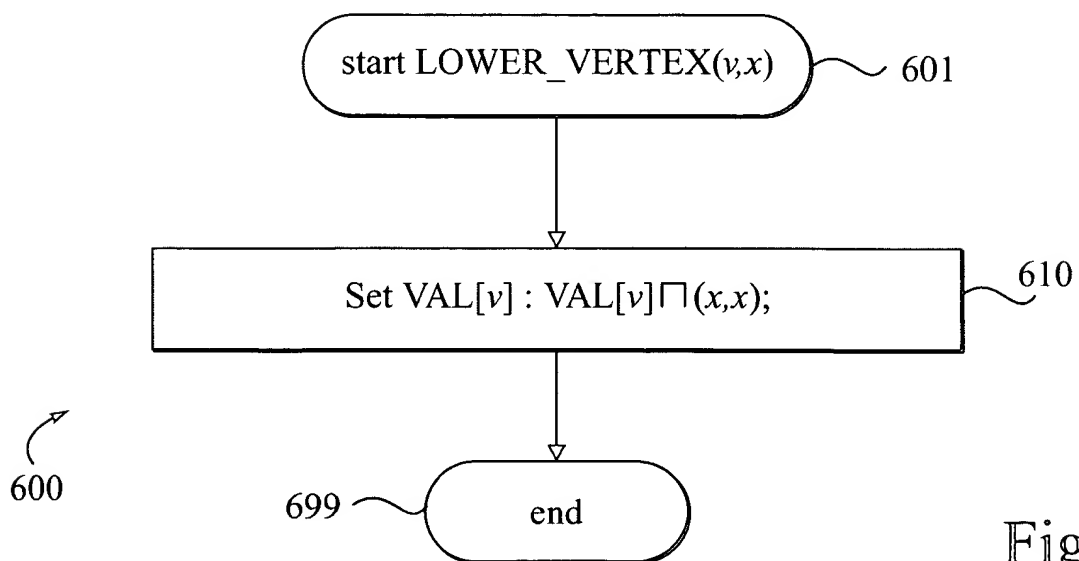
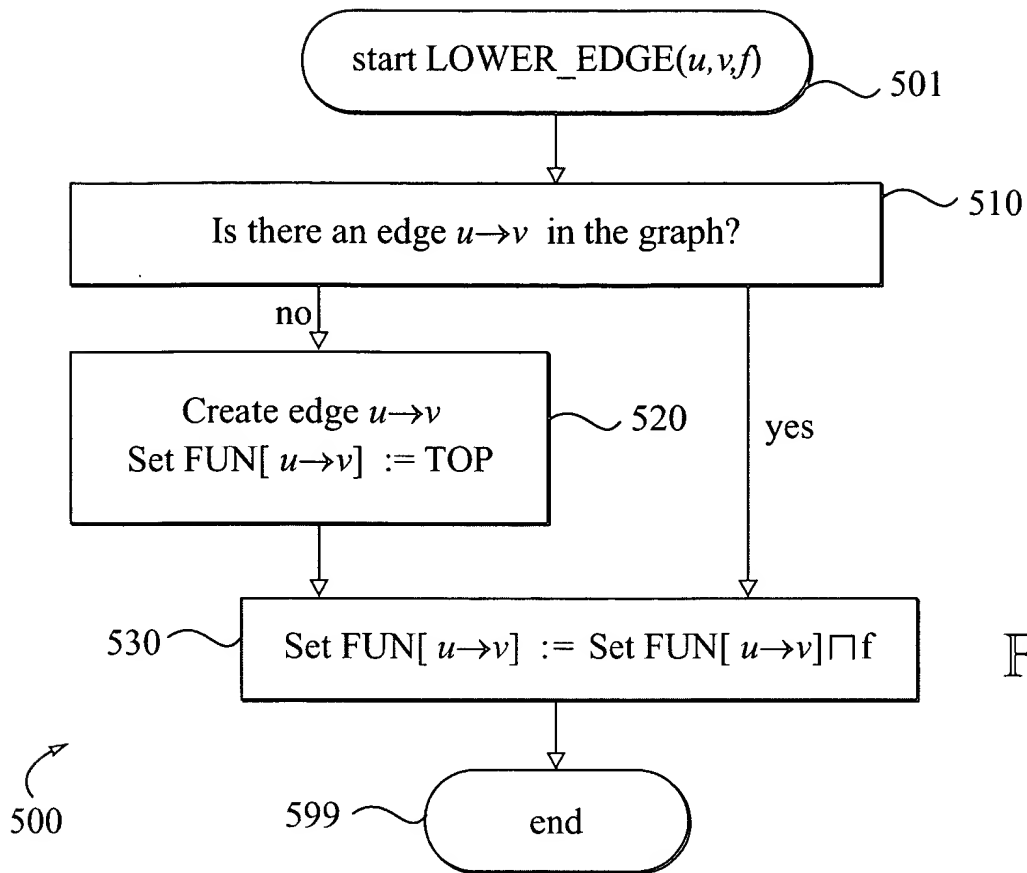


FIG. 4B



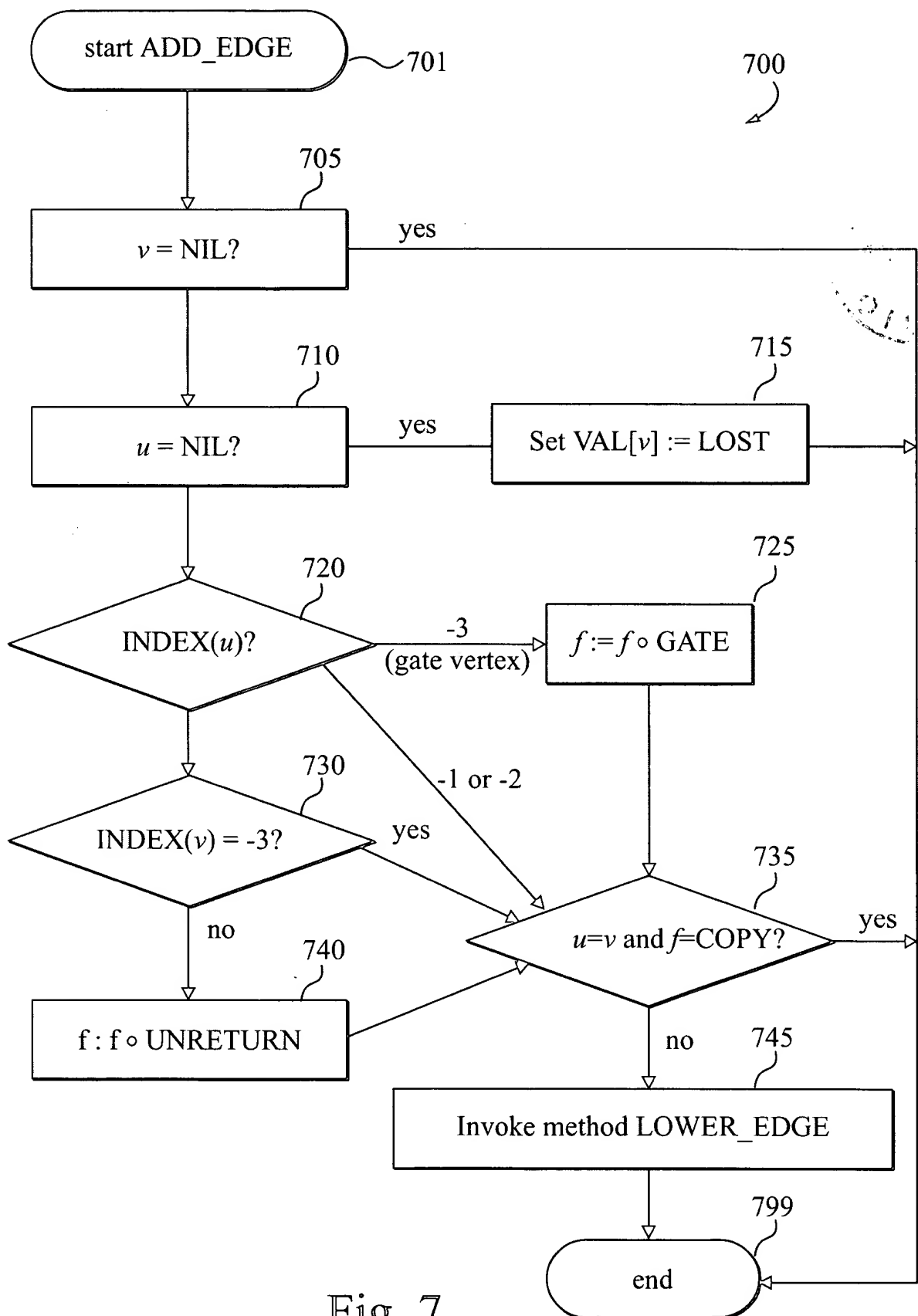


Fig. 7

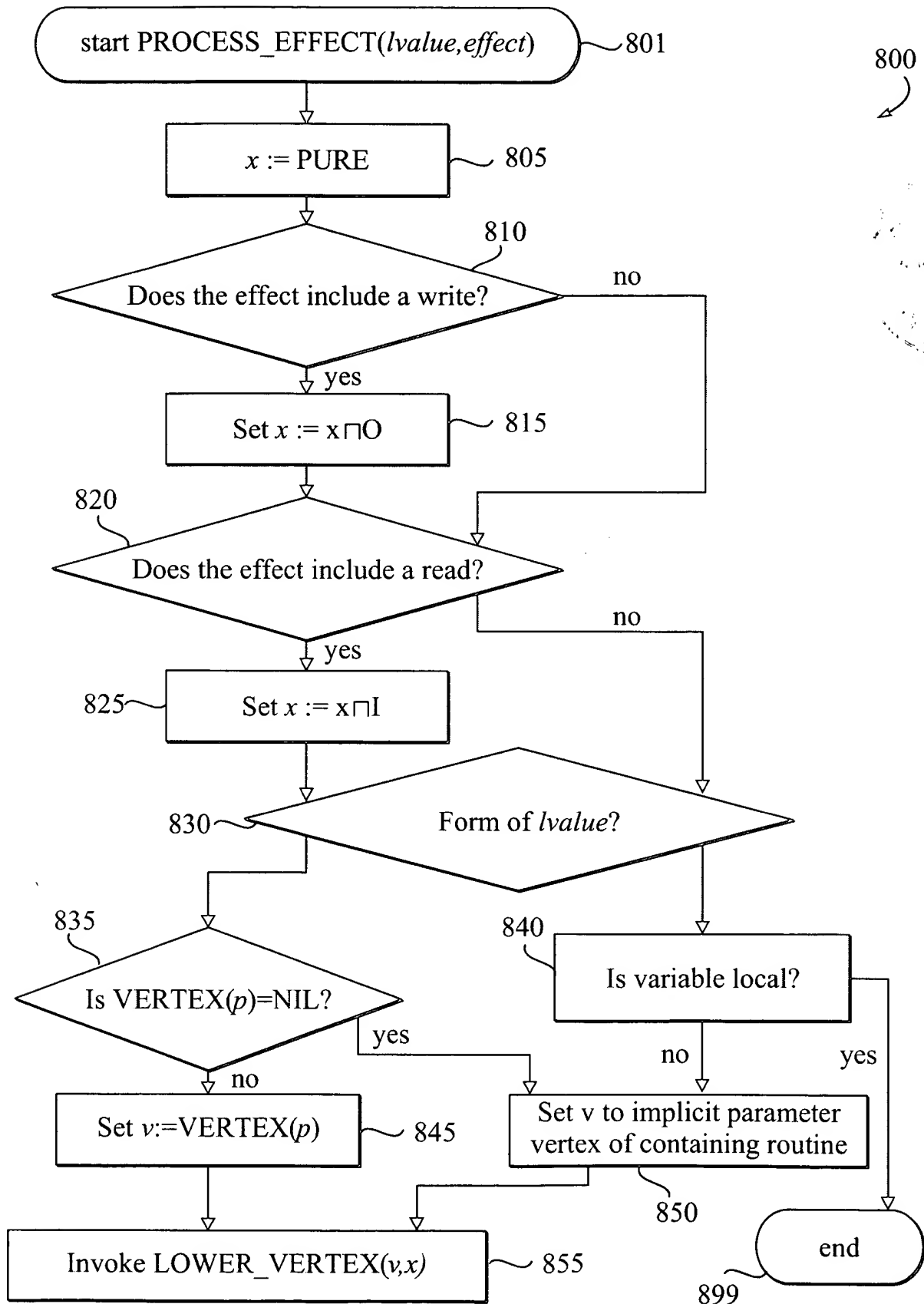


Fig. 8

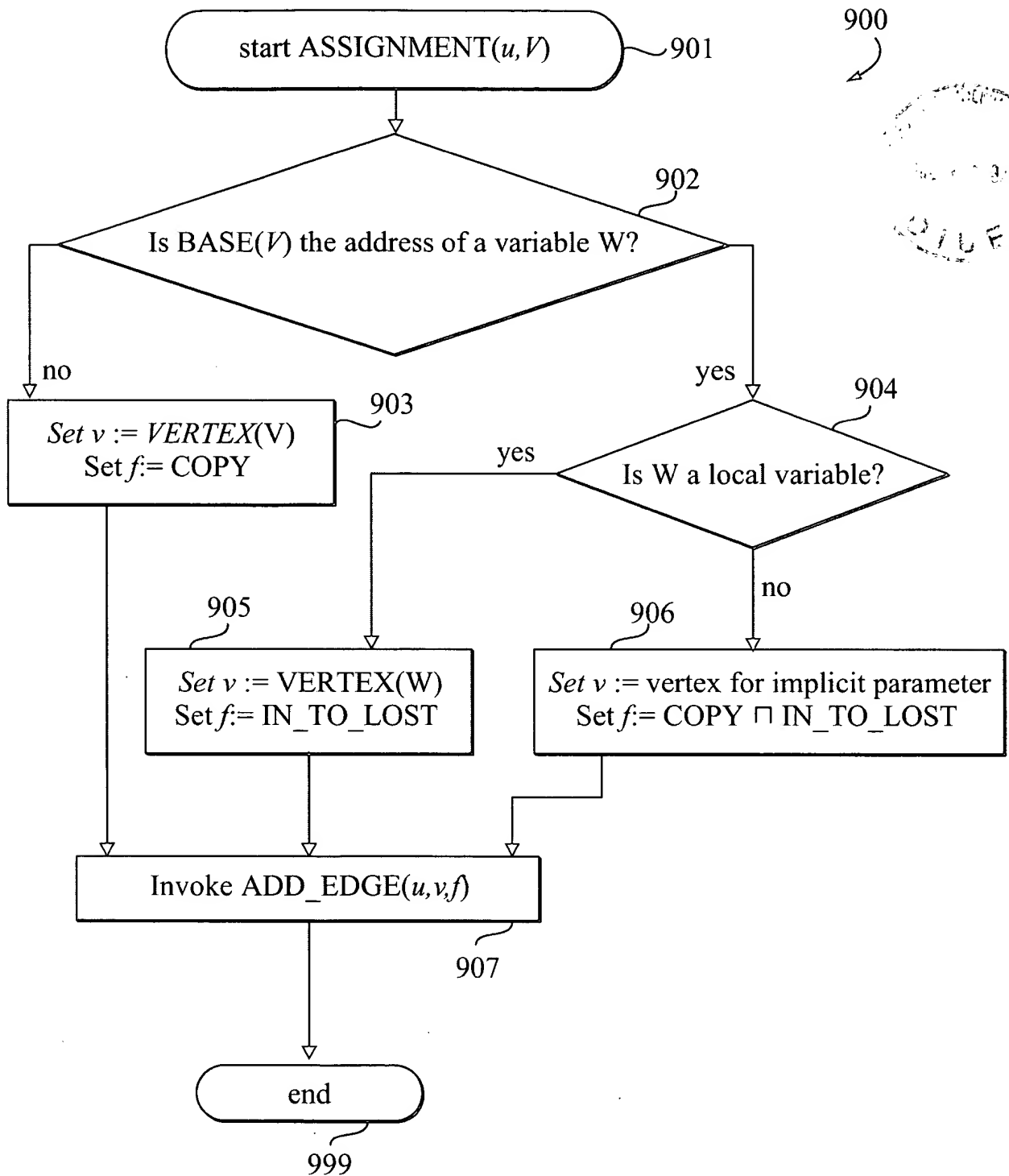


Fig. 9

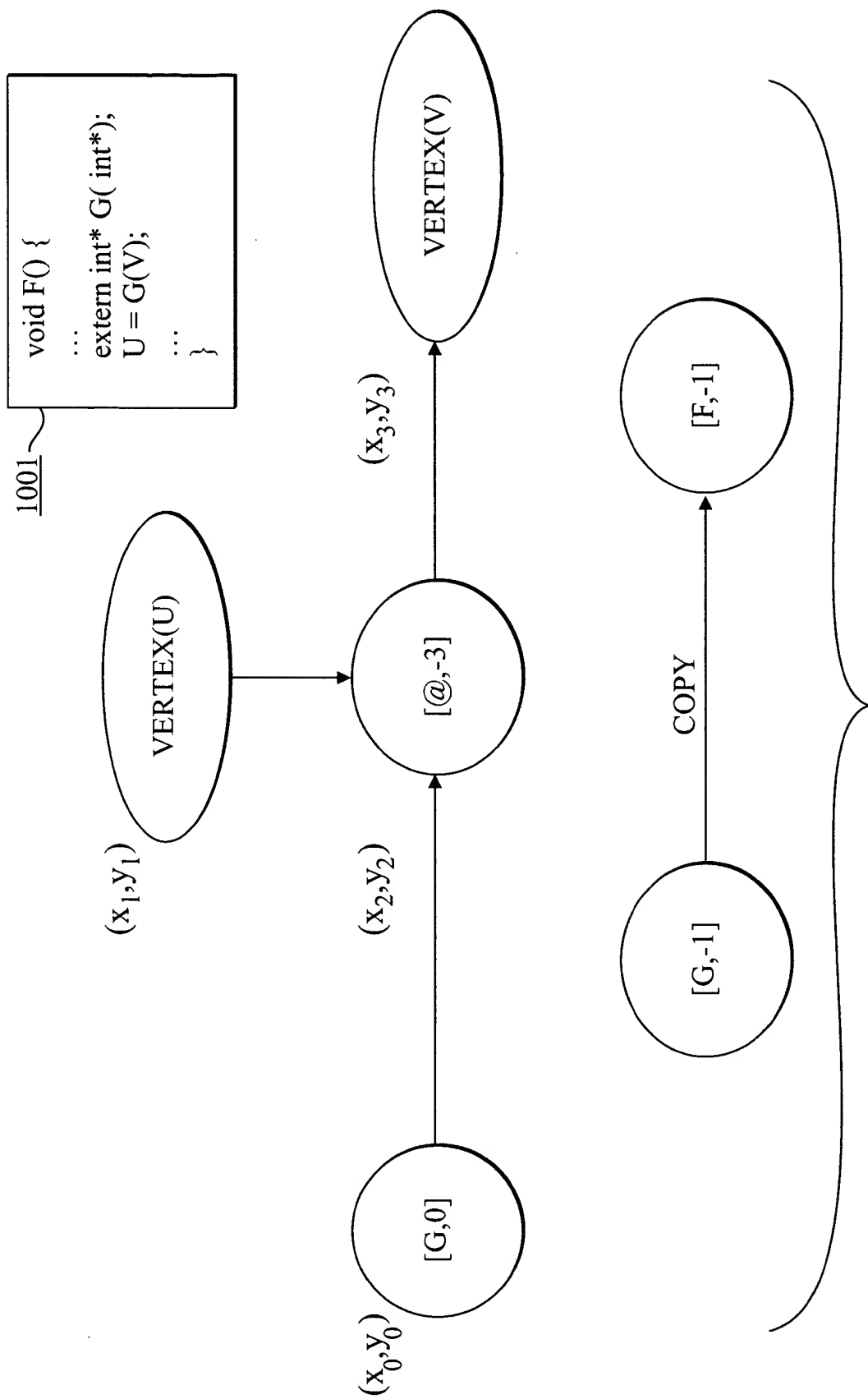


Fig. 10

// Translation unit #1

```
int* f( int* a, int* b, int n ) {  
    int *c = a;  
    if( n>0 ) {  
        int* d = a+1;  
        int* e = b+1;  
        int* z = f( d, e, n-1 );  
        c = z-1;  
        *c = *b;  
    }  
    return c;  
}
```

// Translation unit #2

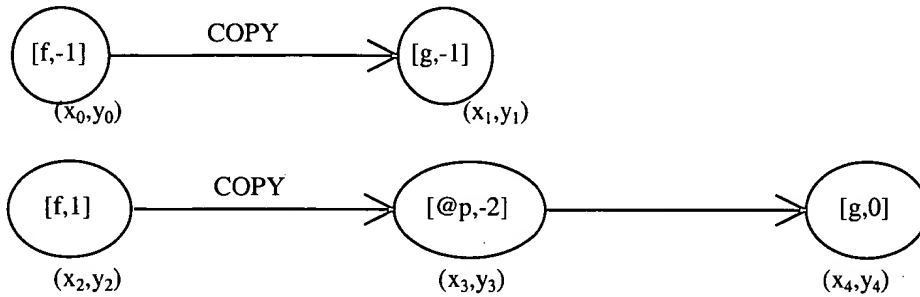
extern int* f(int* a, int* b, int n);

```
void g( int* p ) {  
    int y[10];  
    f( &y[0], p, 10 );  
}
```

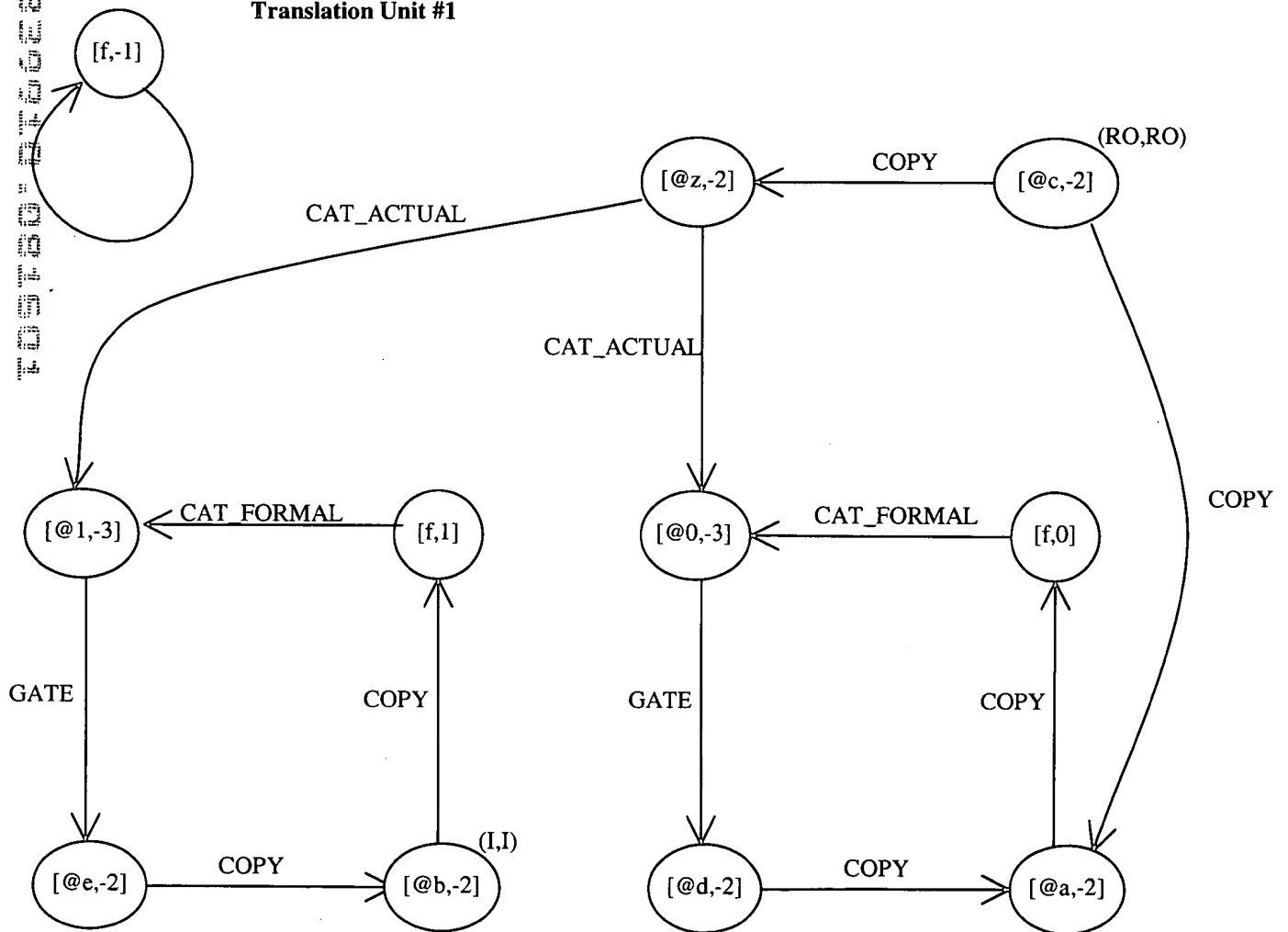
FIG. 11

FIG. 12

Translation Unit #2



Translation Unit #1



Translation Unit #1	
Source line	Action
(entry into f)	Add [$@a,-2$] \rightarrow [$f,0$] Add [$@b,-2$] \rightarrow [$f,1$]
int *c = a;	Add [$@c,-2$] \rightarrow [$@a,-2$]
n>0	None
int *d = a+1;	Add [$@d,-2$] \rightarrow [$@a,-2$]
int *e = b+1;	Add [$@e,-2$] \rightarrow [$@b,-2$]
int* z = f(d,e,n-1)	Add [$@z,-2$] \rightarrow [$@0,-3$] \rightarrow [$@d,-2$] Add [$f,0$] \rightarrow [$@0,3$] Add [$@z,-2$] \rightarrow [$@1,-3$] \rightarrow [$@b,-2$] Add [$f,1$] \rightarrow [$@1,3$] Add [$f,-1$] \rightarrow [$f,-1$]
c = z-1;	Add [$@c,-2$] \rightarrow [$@z,-2$]
*c = *b;	Lower VAL[$[@c,-2]$] to O Lower VAL[$[@b,-2]$] to I
return c;	Lower VAL[$[@c,-2]$] to R

Translation Unit #2	
Action	Action
int *p = &x[0];	None
for(int i=0; i<10; i++)	None (no pointer assignments)
*p = i;	Lower VAL[$[@p,-2]$] to O
p=p+1	None (edge omitted by self-loop rule)
c = z-1;	Add [$@c,-2$] \rightarrow [$@z,-2$]

FIG. 13